Anna Wójtowicz

# Czy maszyny, które ogrywają nas w szachy, myślą?

Studia Philosophiae Christianae 45/1, 57-69

2009

Studia Philosophiae Christianae
UKSW
45(2009)1

ANNA WÓJTOWICZ
*Instytut Filozofii UW, Warszawa*

# CZY MASZYNY, KTÓRE OGRYWAJĄ NAS W SZACHY, MYŚLĄ?

Kiedy uznamy, że maszyna myśli? Albo innymi słowy – kiedy uznamy, że powstała sztuczna inteligencja? Pytanie to obecnie – w czasach dynamicznego rozwoju maszyn zastępujących człowieka w coraz to nowych dyscyplinach – staje się bardzo aktualne. I to nie koniecznie dlatego, że boimy się powstania robotów (które – jak w filmach sf – przejmą władzę nad światem), ale ponieważ odpowiedź na nie pozwala zrozumieć, na czym polega specyfika myślenia człowieka i jakie są cechy charakterystyczne ludzkiej (*vs* sztucznej) inteligencji.

Najogólniej wskazuje się dwa sposoby rozstrzygania problemu, czy maszyna myśli.

Sposób pierwszy polega na tym, że formułujemy pewne obiektywne kryteria uznania danej maszyny za inteligentną[1]. Zgodnie z takim podejściem powiemy, że maszyna myśli, gdy:

– potrafi rozwiązywać określone zadania;
– uczy się;
– wykorzystuje swoje wcześniejsze wyniki, zachowując się w sposób nieprzewidziany przez programistę;
– jest tzw. racjonalnym sprawcą – dostrzega nowe problemy i podejmuje samodzielne działania w celu ich rozwiązania.

Jeśli maszyna będzie posiadała wszystkie te cechy, to uznamy ją za inteligentną, choć nie koniecznie z przydawką „tak, jak człowiek".

[1] Na ten temat por. np. E. Nęcka *Inteligencja*, Gdańskie Wydawnictwo Psychologiczne, Gdańsk 2003.

Sposób drugi sprowadza się do badania, czy maszyna potrafi wykonywać czynności, które uznaje się za wymagające inteligencji, tak, jak robi to człowiek, a więc czy potrafi **symulować działania ludzkiej inteligencji**.

Jako przykłady czynności wymagających inteligencji i pozwalających porównywać maszynę z człowiekiem podaje się tradycyjnie:

- granie w różnego typu gry logiczne;
- dowodzenie twierdzeń w oparciu o przyjęty zbiór aksjomatów i reguł;
- tłumaczenie zdań z jednego języka na inny;
- konwersację w języku naturalnym.

Maszyny radzą sobie z tymi czynnościami w różnym stopniu. Czy jest to stopień wystarczający, aby uznać, że myślą? W sensie pierwszego kryterium muszą sobie z nimi **radzić dobrze, skutecznie** – choć nie do końca jest jasne, co to ma dokładnie znaczyć[2]. Zgodnie z drugim kryterium muszą sobie z nimi **radzić tak, jak robi to człowiek**. Ponieważ jedyną znaną nam inteligencją jest inteligencja ludzka, stąd naturalne a jednocześnie efektywne jest właśnie to drugie podejście.

Klasyczne rozważania na temat (tak rozumianej) sztucznej inteligencji prowadził A. Turing w pracy z 1950 roku: *Computing machinery and intelligence*[3]. Zastąpił on niejasne pytanie: **Czy maszyna myśli?** pytaniem dobrze określonym: **Czy maszyna przejdzie test Turinga?**

Przypomnijmy, że test Turinga polega na grze w udawanie. Wyobraźmy sobie mianowicie, że osoba T (tester) komunikuje się (za pomocą neutralnych – ze względu na cały problem – znaków) z pewną jednostką X. X może być maszyną lub człowiekiem. Te-

---

[2] Np. skonstruowano program, który dowodził twierdzeń zawartych w II rozdziale *Principia Mathematica*. Na 52 główne, udowodnione przez Russella i Whiteheada twierdzenia, maszyna udowodniła 38, w tym jedno ładniejszym i bardziej eleganckim sposobem. Z punktu widzenia kryterium ilościowego maszyna nie poradziła sobie zbyt dobrze, natomiast z punktu widzenia estetycznego – tak. Trudno rozstrzygnąć, które z nich jest istotniejsze ze względu na przypisanie maszynie cechy myślenia.

[3] A. Turing, *Computing machinery and intelligence*, Mind 59(1950), 433–460.

ster zadaje X różne pytania, formułuje zadania i ocenia uzyskane od X odpowiedzi. Uznamy, że X przejdzie test Turinga, jeśli tester T na pytanie, kim jest X odpowie, że X jest człowiekiem.

Innymi słowy, zgodnie z tym kryterium uznamy, że maszyna jest inteligentna, kiedy «oszuka» testera – będzie się zachowywać w sposób nieodróżnialny od tego, jak zachowywałby się człowiek.

Test Turinga jest kryterium operacyjnym. Dla stwierdzenia, czy maszyna jest inteligentna, ważne jest, aby osiągała takie same rezultaty działania w sytuacjach wymagających użycia inteligencji jak człowiek. Nie jest natomiast badana droga, dzięki której maszyna do tych rezultatów dochodzi.

Test Turinga spotkał się z krytyką.

Przede wszystkim podkreśla się wadę testu, jaką jest arbitralność zachowań testera. Objawiać się ona może między innymi tym, że różni testerzy w tych samych okolicznościach będą formułować różne werdykty[4] lub uznawać odpowiedź X za inteligentną, jednocześnie twierdząc, że nie jest to «po ludzku» inteligentne zachowanie (człowiek tak by nie postąpił, zagrał czy odpowiedział[5]). Świadczyłoby to o tym, że niejawnie przyjmują jakieś inne, intuicyjne kryterium inteligencji.

---

[4] Penrose pisze w tym kontekście nawet o testerce, uważając, że kobieta lepiej nadaje się do przeprowadzania przesłuchań. Wyraźniej świadczy to o roli, jaką w teście odgrywa osoba testująca.

[5] W komentarzach do niedawno rozegranego meczu między ówczesnym mistrzem świata Vladimirem Kramnikiem a programem szachowym *Fritz 10* (człowiek przegrał go 2 do 4, nie wygrywając żadnej partii) często pojawiały się stwierdzenia, że komputer gra inteligentne, ale wykonuje «nieludzkie» ruchy. Ciekawe jest jednak to, że – szczególnie wśród młodego pokolenia szachistów – wychowanych na programach szachowych i dużo trenujących z komputerem – ludzie coraz częściej zaczynają grać jak maszyny. Przestają się kierować pewnymi ogólnymi cechami pozycji (jej estetyką), ale zaczynają analizować «nienaturalne» warianty. Dążą do pozycji, które dla człowieka wyglądają źle, ale które komputer – w oparciu o brutalne wyliczenie – ocenia jako dobre. Chciałoby się powiedzieć, że z punktu widzenia ludzkiego, są one skuteczne z czysto przypadkowych powodów. To pokazuje, że nie tylko komputery starają się naśladować ludzi, ale również ludzie zaczynają działać jak komputery – co może zaburzać wyniki testu Turinga.

Niektórzy (np. Minsky) uznają je za zbyt silne kryterium, bo inteligencję powinno się przypisywać właściwie każdemu urządzeniu, które ma wbudowane sprzężenie zwrotne (modyfikuje swoje zachowania wraz ze zmianą otaczających go warunków). Przy takim podejściu inteligencję należałoby przypisać np. termostatowi. Inni z kolei uznają test Turinga za kryterium zbyt słabe – nie można urządzeniu, które na poziomie czysto syntaktycznym manipuluje symbolami przypisywać cechy rozumienia tych symboli – a właśnie rozumienie i intencjonalność są podstawą inteligencji. Takiej tezy broni np. Searle, konstruując swój słynny argument chińskiego pokoju. Ten argument wymaga szczególnej uwagi i wrócimy do niego za chwilę.

Jeśli akceptujemy test Turinga, to powyższe zarzuty wymagają odpowiedzi.

Moim zdaniem – i takiej tezy chciałabym bronić w tym artykule – kluczowe dla oceny testu Turinga, jako skutecznej metody ustalania, czy X myśli, jest wybór sytuacji testowych, czyli wybór tych zadań, które tester T będzie stawiał przez X, aby stwierdzić, czy ma do czynienia z maszyną, czy z człowiekiem.

Zacznijmy nasze rozważania od następującej obserwacji. Zauważmy mianowicie, że często to samo zadanie można rozwiązać na kilka sposobów, a sposoby te można porządkować ze względu na to, w jakim stopniu wykorzystują one inteligencję.

Zilustrujmy to trzema przykładami:

1) Zadanie polega na zsumowaniu liczb od 1 do 100.

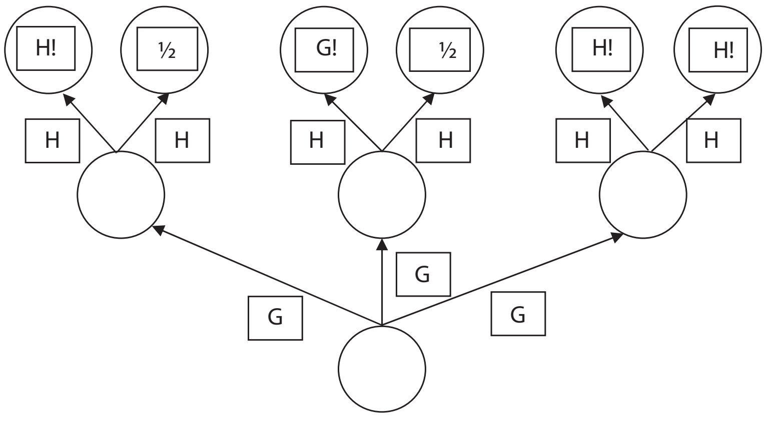
Osoba A uzyskuje wynik przez zwykłe sumowanie kolejnych liczb (1 + 2 +..+ 100).

Osoba B zauważa, że suma par liczb skrajnych w tym ciągu wynosi zawsze 101 (1 + 100, 2 + 99..50 + 51), i par takich liczb jest 50. Uzyskuje wynik przez mnożenie 101 razy 50.

Metoda druga – wymyślona przez młodego Gaussa – jest inteligentniejsza (bardziej pomysłowa) niż pierwsza.

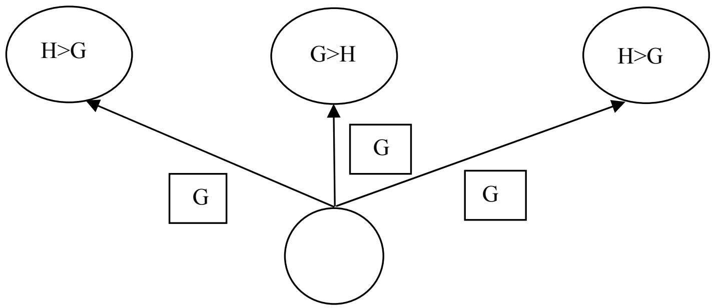
Zanim przejdziemy do następnego przykładu, zróbmy krótką dygresję dotyczącą gier strategicznych i architektury programów, zgodnie z którymi działają maszyny grające w takie gry.

Rozważmy dowolną strategiczną grę dwuosobową dla graczy G i H. Z logicznego punktu widzenia każda pozycja (każda sytuacja) w tej grze stanowi początek pewnego **drzewa wariantów** (rys.):




Ilość gałęzi drzewa na każdym poziomie zależy od ilości teoretycznie możliwych posunięć, które gracz może wykonać w danej pozycji (np. w szachach na każdy pierwszy ruch białych – takich ruchów jest 20 – czarne mogą wybrać którąś z 20 możliwych odpowiedzi). Drzewo dość szybko się rozrasta. Jego wierzchołki stanowią pozycje, które z punktu widzenia zasad danej gry mają jednoznaczną ocenę (w których wygrał gracz G – oznaczamy to na rysunku przez **G!**, wygrał gracz – oznaczamy to przez **H!** lub uzyskano remis – oznaczamy to przez **1/2**). Z teoretycznego punktu widzenia, w każdej pozycji wybór posunięcia przez gracza powinien następować zgodnie z zasadą minimaksu – należy wybrać takie odgałęzienie, w którego wierzchołkach znajdują się wyniki najkorzystniejsze dla wybierającego. (Na naszym rysunku gracz G powinien wybrać wariant środkowy, ponieważ w najgorszym przypadku uzyska remis. Warianty lewy i prawy mogą doprowadzić do jego przegranej).

Wszystkie drzewa wariantów (nawet pełne – tzn. zaczynające się od pozycji wyjściowej danej gry) dla znanych gier są skończone (gwarantują to najczęściej reguły gry). Dla gier prostych (np. kółko i krzyżyk) drzewa takie są proste i możliwe do pełnego prześledzenia. Dla gier złożonych (szachy, go) stworzenie pełnego drzewa jest bardzo trudne lub praktycznie niemożliwe. Dlatego konstruuje się drzewa częściowe. Idea polega na tym, aby redukować drzewo przez obcinanie gałęzi na podstawie **funkcji oceniającej**. (rys.):




Funkcja ta przypisuje pozycji, która z punktu widzenia zasad danej gry nie jest jeszcze rozstrzygnięta, określoną wagę. Innymi słowy, gracz ocenia odgałęzienie nie na tej zasadzie, że zna wszystkie jego konsekwencje (czyli wszystkie wierzchołki drzewa), ale dlatego, że funkcja oceniająca przypisała mu wyższą wagę. Po tym następuje wybór posunięcia na podstawie zasady minimaksu.

Jakość programu grającego w daną grę (czyli programu zastępującego jednego z graczy) zależy w związku z tym od dwóch czynników: od możliwości obliczeniowych (czyli od tego, jak duże drzewo program potrafi narysować) i od jakości funkcji oceniającej – na ile trafnie potrafi ona przyporządkować każdej pozycji wagę (czy drzewo zredukowane jest dobrym uproszczeniem drzewa pełnego). W skrajnych wypadkach można mieć idealną funkcję oceniającą i maksymalnie zredukowane drzewo lub pełne drzewo bez żadnej funkcji.

Zauważmy również w tym miejscu, że zasadnicza różnica między człowiekiem a maszyną grającą w daną grę polega na tym, że człowiek w dużo większym stopniu polega na funkcji oceniającej, co zwalnia go z konieczności budowania pełnego drzewa wariantów. Pewne jego gałęzie odcina od razu jako absurdalne.

2) Rozważmy teraz dwuosobową grę strategiczną (np. warcaby), w którą z pewnym mistrzem grają równolegle gracze A i B.

Niech gracz A dysponuje pełnym drzewem rozważanej przez nas gry. Jego strategia polega na wędrówce po tym drzewie zgodnie z zasadą minimaksu. Na ruch mistrza, gracz A wybiera taką odpowiedź, która prowadzi w drzewie wariantów do korzystnego dla niego re-

zultatu. Zauważmy, że gracz A, aby wygrać z mistrzem, nie musi znać nawet reguł gry w warcaby – wystarczy, aby umiał posługiwać się drzewem wariantów i zasadą minimaksu.

Załóżmy, że gracz B nie dysponuje drzewem gry i musi na bieżąco wymyślać posunięcia w odpowiedzi na ruchy mistrza.

Jeśli rezultaty graczy A i B będą takie same (tzn. jeśli gracz B faktycznie dobrze gra w warcaby), to trzeba się zgodzić, że gracz B musiał się wykazać większą inteligencją niż gracz A. Mówiąc w skrócie, jest tak dlatego, że tworzenie (w umyśle) drzewa wariantów wymaga większej inteligencji niż wędrowanie po zadanym wcześniej drzewie zgodnie z zasadą minimaksu.

3) Jako kolejny przykład rozważmy problem wzorowany na chińskim pokoju Searle'a[6]. Załóżmy, że w zamkniętym pokoju jest umieszczona osoba A. Do pokoju dostarczane są pytania sformułowane w języku chińskim. Osoba A nie zna języka chińskiego, ale ma do dyspozycji bardzo dobre instrukcje napisane w zrozumiałym dla siebie języku (np. w języku polskim), które pozwalają jej – poprzez zgodne z tymi instrukcjami manipulacje symbolami chińskimi – produkować sensowne odpowiedzi w języku chińskim na zadawane pytania. To, że osoba A z punktu widzenia zewnętrznego obserwatora prowadzi konwersację po chińsku, nie zmienia faktu, że osoba A nie rozumie chińskiego, a więc sensu zadawanych pytań i tworzonych przez siebie odpowiedzi.

Dla potrzeb naszego przykładu załóżmy, że w pokoju obok umieszczona jest osoba B, która zna dobrze język chiński. Dostaje ona takie same pytania jak osoba A i – po namyśle – udziela na nie dobrych (takich samych, jak osoba A) odpowiedzi.

Podobnie jak w przykładzie poprzednim, uznamy, że osoba B, samodzielnie znajdująca odpowiedzi na pytania, musi wykazywać się większą inteligencją niż osoba A, od której wymagana jest jedynie umiejętność wypełniania instrukcji. Zakładamy bowiem, że udzielanie właściwych odpowiedzi na pytania jest trudniejsze niż realizacja instrukcji.

Podsumowując: w powyższych przykładach mieliśmy do czynienia z zadaniami, które można było rozwiązać na dwa sposoby, przy

---

[6] Por. J. R. Searle, *Minds, brains and programs*, The Behavioral and Brain Sciences 3(1980), 417–451.

czym realizacja sposobów wybieranych przez osobę A wymagała mniejszej inteligencji niż realizacja sposobów wybieranych przez osobę B. Jeśli nie jesteśmy zwolennikami tezy, zgodnie z którą inteligencję przypisujemy termostatowi (a mam nadzieję, że nie jesteśmy), to maszynę, która rozwiązuje zadania w taki mało sprytny sposób (sumuje po kolei liczby, wędruje po drzewie wariantów zgodnie z zasadą minimaksu, czy mechanicznie realizuje proste instrukcje), jak robiła to osoba A w naszych przykładach, w ogóle nie uznalibyśmy za inteligentną.

Co by się jednak stało, gdyby zadania te stanowiły sytuacje testowe w teście Turinga? Pamiętajmy, że test ten ma ściśle operacyjny charakter. Nie pozwala w związku z tym zaglądać w architekturę działania testowanej jednostki (czyli badać, w jaki sposób rozwiązuje ona zadanie), a jedynie ocenia ją po wynikach. Nie wiedzielibyśmy, czy X rozwiązuje te zadania jak osoba B i jest inteligentny, czy też jak osoba A i inteligencją się w zasadzie nie posługuje. Samo przejście testu Turinga (rozwiązanie zadań) nie wystarczałoby do uznania X za jednostkę inteligentną. Oznacza to, że zadania (1)–(4) nie mogą się znaleźć w teście Turinga, ponieważ test ten byłby wtedy – tak jak twierdzi Searl – zbyt słaby.

Aby obronić test Turinga, należy się w związku z tym zastanowić, jakie warunki musi spełniać sytuacja testowa, aby można ją było nazwać turingowską, tzn. aby jej wystąpienie w kryterium operacyjnym zapewniało jego skuteczność – pozwalało odróżnić maszyny inteligentne od nieinteligentnych?

Uważam, że sytuacja turingowska jest to sytuacja spełniające następujące trzy warunki:

Po pierwsze, musi istnieć jasne, jednoznaczne kryterium tego, że postawiony problem został rozwiązany tak, jak zrobiłby to inteligentny człowiek. W tym sensie konkurs pisania wierszy czy konwersacja na dowolne tematy nie są optymalnym przykładem sytuacji turingowskiej. Najlepiej oczywiście, gdyby sytuacja testowa była grą o ściśle zadanych regułach i ustalonej puli wyników (wygrana, przegrana, remis). W ten sposób zminimalizowana byłaby rola testera – nie musielibyśmy polegać na **subiektywnym** poczuciu testera, że X jest bądź nie jest – maszyną.

Po drugie, w sytuacji turingowskiej warunkiem wystarczającym rozwiązania danego zadania musi być użycie inteligencji. Taki wymóg można rozumieć na kilka sposobów:

– przy rozumieniu najmocniejszym żądamy, aby rozwiązanie danego zadania **nie miało charakteru losowego** (odpadają pytania typu – w jednym z pudełek jest kulka – zgadnij w którym);

– przy rozumieniu najsłabszym rozwiązanie zadania **musi być przynajmniej teoretycznie algorytmiczne**. To gwarantuje (na mocy tezy Churcha), że problem jest w ogóle rozwiązywalny przez maszyny Turinga

Jeśli przyjmiemy pierwszą z tych charakterystyk, to w szczególności rozwiązanie zadania może mieć charakter niealgorytmiczny. Wynika stąd, że nie rozwiążą go maszyny mające strukturę maszyn Turinga. Nie oznacza to jednak a *priori*, że żadna maszyna nie potrafi sobie z nim poradzić. Problemy niealgorytmiczne mogą być rozwiązywane np. przez tzw. sieci neuronowe. Samo nałożenie takiego warunku na sytuacje turingowskie nie przesądza więc problemu, czy maszyny są w stanie przejść ten test i nie czynią kryterium Turinga zbyt silnym.

Po trzecie i najważniejsze, warunkiem koniecznym rozwiązania danego zadania musi być użycie inteligencji. Rozwiązanie problemu nie sprowadza się do wykorzystania brutalnej siły obliczeniowej (wykonania prostych, czysto mechanicznych przekształceń). W tym sensie sytuacją turingowską nie jest wykonanie mnożenia czy gra w warcaby – ogólnie gra, dla której mamy dostępne pełne drzewo wariantów. Chodzenie po tym drzewie nie wymaga inteligencji w takim stopniu, jaki nas interesuje. Oznacza to innymi słowy, że dany problem jest dobry jako sytuacja turingowska, jeśli jest **praktycznie niealgorytmiczny.**

Z przeprowadzonych analiz można uzyskać następujące wnioski:

1) Wniosek ogólnometodologiczny.

Kryterium operacyjne (takie jak test Turinga) pozwalające stwierdzić, że X ma własność C, jest dobre tylko wtedy, gdy czynności X–a, których rezultaty oceniamy, rzeczywiście angażują tylko i wyłącznie badaną własność C. To źle dobrana sytuacja testowa powoduje, że test może być:

– niemiarodajny, gdy jego wyniki – np. ze względu na osobę testera – są niejednoznaczne;

– za silny, gdy aby go przejść, niezbędna jest jakaś dodatkowa umiejętność poza C – w przypadku testu Turinga np. intuicja;

– za słaby, gdy aby wykonać zadanie, własność C nie jest konieczne potrzebna – w przypadku testu Turinga może np. wystarczać wiedza na temat drzewa wariantów.

Oczywiście powstaje problem, skąd mamy wiedzieć, że dana czynność angażuje tylko i wyłącznie własność C. W teście Turinga nie możemy zaglądać do wnętrza komputera, badać architektury programu – i jeśli zobaczymy, że działa on prymitywnie, to odrzucać rozwiązywane przez niego zadania jako sytuację nieturingowską. Oznaczałoby to bowiem, że posługujemy się nie tyle testem Turinga do stwierdzenia, czy dana jednostka jest inteligentna, ale kryterium budowy działającego programu – jeśli jest on dostatecznie prosty, to nie uznamy realizującej go maszyny za inteligentną.

2) Wniosek dotyczący argumentacji Searla.

Według Searla, eksperyment myślowy, którym posłużyliśmy się w opisie osoby A z przykładu (3), ma pokazywać, że samo manipulowanie symbolami zgodnie z pewnym narzuconym zbiorem reguł, a więc na poziomie czysto syntaktycznym, nie dowodzi myślenia i inteligencji. Zauważmy w tym miejscu, że istotnym założeniem, które leży u podstaw argumentu chińskiego pokoju jest to, że istnieje (aktualnie) drzewo konwersacji dla języka chińskiego. Tylko bowiem w takim wypadku instrukcje, które posiada osoba A, rzeczywiście będą działały skutecznie. Przyjęcie tego założenia oznacza, że problem konwersacji jest problemem analogicznym do omówionej w przykładzie (2) gry w warcaby – jego rozwiązanie nie wymaga inteligencji w sposób konieczny (podobną uwagę na ten temat można znaleźć u N. Blocka i we wcześniejszej pracy S. Lema[7]). Jest to zgodne również z naszymi obserwacjami – na tej samej podstawie w przykładzie dotyczącym gry w warcaby, uznajemy zachowanie osoby A za wymagające mniej inteligencji niż zachowanie osoby B. Searle pokazał więc, że przy założeniu, że konwersacja w języku chińskim nie jest sytuacją turingowską, zastosowanie jej w teście Turinga spowoduje, że będzie on zbyt słabym kryterium inteligencji. Na to oczywiście zgoda z definicji, ale nie jest to wniosek ciekawy ani groźny dla testu Turinga.

[7] Por. S. Lem, *Summa technologiae*, Interart, Warszawa 1996 (rok napisania 1964).

Searle wyciąga stąd jednak dalsze dwa wnioski. Twierdzi on, że **żadnej** maszynie (manipulującej z założenia symbolami) nawet, gdyby przeszła ona test Turinga, nie można przypisać rozumienia tych symboli, a co za tym idzie, również inteligencji. W naszej terminologii oznacza to tyle, że, według Searle'a, po pierwsze, jedynym sposobem, w jaki maszyna może sobie poradzić z graniem w grę logiczną lub prowadzeniem konwersacji jest – z definicji wręcz – posłużenie się pełnym drzewem wariantów dla danego problemu, a po drugie każda sytuacja, która pojawi się w teście Turinga musi być nieturingowska.

Pierwszy wniosek Searle'a ma charakter deklaracji i w tym sensie nie jest interesujący. Aby obalić całe rozumowanie Searle'a, należy wykazać, że istnieją sytuacje turingowskie. Jeśli sytuacje te wykorzystamy w teście Turinga, to będzie to test adekwatny. Jeśli jakaś maszyna przejdzie taki test Turinga, to będziemy musieli uznać ją za inteligentną.

3) Wniosek dotyczący klasy sytuacji turingowskich.

Wraz z rozwojem maszyn i gromadzeniem wiedzy, coraz więcej czynności, które do tej pory wydawały się wymagać inteligencji w sposób konieczny i być przykładem dobrej sytuacji turingowskiej, przestają takie być, bo do ich rozwiązania nie wymagana jest inteligencja, ale zgromadzona informacja i siła obliczeniowa. Jeśli słusznie oszacowaliśmy, że sytuacje turingowskie muszą być teoretycznie, ale nie praktycznie algorytmiczne, to wynika to stąd, że zmienia się w czasie pojęcie praktycznej obliczalności. Sytuacje uznawane za turingowskie przez samego Turinga (np. gra w szachy) obecnie takie już nie są. Niektóre problemy zostały już bowiem bezpowrotnie rozwiązane. Wiemy, że aby w takich sytuacjach postępować tak, jak inteligentny człowiek, komputer nie musi myśleć – potrzebuje tylko informacji. W szczególności, ze względu na zgromadzoną bazę danych gra w szachy nie jest już sytuacją turingowską. Stąd też negatywna odpowiedź na pytanie postawione w tytule artykułu.

4) Wniosek dotyczący wykonywalności testu Turinga.

Powstaje pytanie, czy test Turinga jest w ogóle wykonalny, tzn. czy istnieją sytuacje turingowskie we wskazanym wyżej sensie. Według mnie, dobrymi sytuacjami turingowskimi spełniającymi nakładane na nie trzy warunki są gry strategiczne o tak dużej złożoności, że skonstruowanie dla nich drzewa wariantów jest praktycznie (w tej

chwili) niewykonalne. Niektórzy mogą za taką specjalną grę uważać np. konwersację. Jej wadą jest to, że w ustaleniu wyniku testu konwersacyjnego bardzo dużo zależy od osoby testera. Ja sądzę, że lepszym przykładem sytuacji turingowskiej jest gra w go. Ma ona również tę zaletę (i przewagę np. nad szachami), że w sposób naturalny można zwiększać stopień jej komplikacji bez zmiany zasadniczych reguł samej gry. Wystarczy powiększyć planszę, na której gra się odbywa. Nie powinno to wpłynąć istotnie na siłę inteligentnego człowieka, który i tak posługuje się funkcją oceniającą, pozwalającą mu obcinać gałęzie w drzewie wariantów. Będzie natomiast zabójcze dla komputera, który rozwiązuje problem siłowo. Dzięki zwiększaniu planszy mamy gwarancję, że – poczynając od planszy pewnej wielkości – na pewno nie istnieje (obecnie) drzewo wariantów dla takiej gry.

Test Turinga z grą w go wyobrażam sobie więc następująco: tester T rozgrywa z X kilka partii w go, za każdym razem powiększając planszę. Jeżeli siła gry X nie zmieni się istotnie, to tester powinien uznać X za człowieka. Gdyby jednak X był maszyną, to będziemy musieli – moim zdaniem – zgodzić się, że X jest maszyną myślącą i że powstała sztuczna inteligencja.

I na koniec zapowiedziany we wstępnie wniosek dotyczący ludzkiej inteligencji. Z przytoczonych przykładów wynika, że inteligentne rozwiązanie danego problemu – nawet takiego, który teoretycznie można rozwiązać siłowo – polega na tym, że znajdujemy drogę na skróty: szybciej dodajemy liczby, redukujemy drzewa wariantów. Możemy pokusić się w związku z tym o następującą (daleką od standardowej) definicję inteligencji:

Inteligencja to własność umysłu pozwalająca przy rozwiązywaniu problemów unikać komplikacji obliczeniowych.

**THE MACHINES THAT BEAT US AT CHESS – DO THEY THINK?**

Summary

The article is devoted to an analysis of some counterarguments to Turing's test. It is argued, that the proper choice of test situations is important for the discussion of Turing's test. Three criteria of being a Turing situation are formulated and discussed.